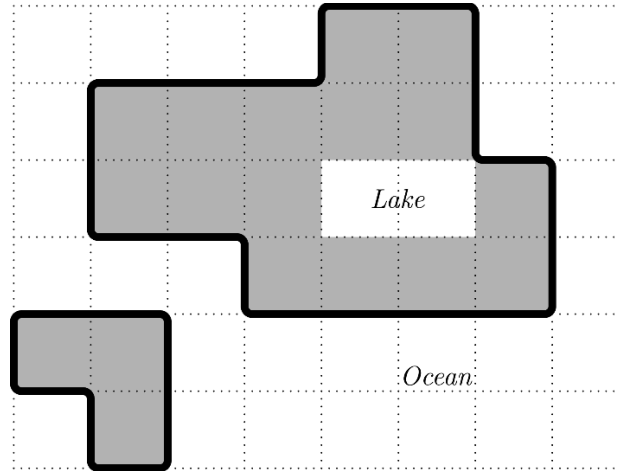


8. Erbatur

Program Name: Erbatur.java

Input File: erbatur.dat

Erbatur loves spending his summers in the Bahamas but has always struggled to pick which of the islands in the Bahamas to stay at. One factor that is important to Erbatur is how large of an island he is staying on. More specifically, the length of the coastline (where the ocean meets the land of the island) plays a large factor in how much Erbatur enjoys where he stays. Knowing that planning trips is a large undertaking, rather than look at the entire map of the Bahamas, Erbatur decides to look at only a subsection of the map and wants to know how long the total coastline is of all islands contained within his view. Help Erbatur write a program that automates this process.



Gray squares represent land, solid black lines outline the coastline of two disjoint islands.

Input: The first line of input will consist of a single integer, T ($1 \leq T \leq 10^2$), denoting the number of test cases to follow. Each test case will begin with a single line of two space-separated integers n and m ($1 \leq n, m \leq 10^3$) denoting the size of the $n \times m$ map-view to follow. The next n lines will each consist of m characters, each of which is either a '.' to represent water, or a '#' to represent land. You may assume that the map-view of each test case is surrounded by ocean.

Output: For each of Erbatur's T queries, each on their own line, output a single integer ℓ_i denoting the total length of the coastline of all islands, the i^{th} of which corresponds to Erbatur's i^{th} query. Note that water that appears within a given map view is not guaranteed to be a part of the ocean. For a cell of water to be considered a part of the ocean, it must be connected to the border of the map only through other cells of water.

Sample input:

```
5
6 8
.....##..
.#####.
.###...#.
...####.
##.....
.#.....
.#.....
5 6
.####.
.#.##.
###...
....#
.....
```

~ Input continues next column ~

~ Input continued ~

```
3 2
.#
..
#.
5 10
.....#.....#
...#.....##
..#.....##.
.#.....##..
.#.....##..
#.....##...
#.....##...
3 5
.....
.....
.....
```

Sample output:

```
28
20
8
40
0
```